

We claim:

1. A computer-implemented method of enhancing cache performance, the method comprising:
 - receiving a temporal data reference profile;
 - detecting one or more hot data streams in the temporal data reference profile; and
 - analyzing the one or more hot data streams and the temporal data reference profile to determine a coallocation solution for allocations in heap memory.
2. The method of claim 1 wherein the temporal data reference profile traces accesses of objects, and wherein the one or more hot data streams are for object accesses.
3. The method of claim 1 wherein the temporal data reference profile traces accesses of object fields, and wherein the one or more hot data streams are for object field accesses.
4. The method of claim 1 wherein the coallocation solution includes one or more of field reordering, object splitting, and object merging.
5. The method of claim 1 further comprising:
 - altering a computer program to implement the coallocation solution.
6. The method of claim 5 wherein the altering comprises binary rewriting of a computer program.
7. The method of claim 5 wherein the altering comprises rewriting of part of source code of the computer program under control of a programmer.
8. The method of claim 5 wherein the altering comprises adjusting executable code for the computer program at compile time.

9. The method of claim 5 further comprising:
after the altering, executing the computer program, wherein run time support software enforces the coallocation solution.

5 10. A computer-readable medium storing the computer program altered according to the method of claim 5.

11. The method of claim 1 wherein the coallocation solution is based at least in part on results of weighted set packing analysis for plural coallocation sets.

10 12. The method of claim 11 wherein each of the plural coallocation sets comprises a set of allocation sites for objects in one of the hot data streams.

15 13. The method of claim 11 wherein each of the plural coallocation sets comprises a set of allocation sites for object fields in one of the hot data streams.

14. The method of claim 1 wherein the analyzing comprises:
computing a heat value for each of the one or more hot data streams; and
computing a weight value for each of the one or more hot data streams.

20 15. The method of claim 14 wherein the analyzing further comprises:
avoiding double counting for sub-sets among the one or more hot data streams.

25 16. A computer-readable medium storing computer-executable instructions for causing a computer system programmed thereby to perform the method of claim 1.

17. A computer-readable medium storing computer-executable instructions for causing a computer system programmed thereby to perform a method of enhancing cache performance, the method comprising:

30 receiving a profile of object field accesses; and

determining a coallocation solution based at least in part upon the profile, wherein the coallocation solution increases locality of object fields in a layout in memory to improve cache performance.

5 18. The computer-readable medium of claim 17 wherein the profile is a temporal data reference profile that traces the object field accesses.

19. The computer-readable medium of claim 17 wherein the method further comprises:

10 detecting one or more hot data streams in the profile.

20. The computer-readable medium of claim 17 wherein the coallocation solution includes object restructuring in a computer program, the method further comprising:

15 executing the computer program after the object restructuring, wherein the coallocation solution guides object-level allocations in heap memory.

21. The computer-readable medium of claim 20 wherein the object restructuring includes field reordering.

20 22. The computer-readable medium of claim 20 wherein the object restructuring includes object splitting.

23. The computer-readable medium of claim 20 wherein the object restructuring includes object merging.

25 24. The computer-readable medium of claim 17 wherein the coallocation solution includes object restructuring in a computer program, the method further comprising:
 performing the object restructuring at design time.

25. A computer-readable medium storing the computer program altered according to the method of claim 24.

26. The computer-readable medium of claim 17 wherein the coallocation solution
5 includes object restructuring in a computer program, the method further comprising:
performing the object restructuring at run time.

27. The computer-readable medium of claim 17 wherein the method further comprises:

10 executing a computer program, wherein the coallocation solution guides field-level allocations in heap memory.

28. The computer-readable medium of claim 17 wherein the determining comprises:

15 computing a weighted set packing for one or more field coallocation sets; and selecting the coallocation solution.

29. A computer-readable medium storing computer-executable instructions for causing a computer system programmed thereby to perform a method of enhancing cache
20 performance, the method comprising:

receiving a comprehensive temporal data access profile for a computer program;
detecting one or more data access patterns in the temporal data access profile,
each of the one or more data access patterns including plural data accesses and occurring
one or more times in the temporal data access profile; and
25 analyzing the one or more data access patterns and the temporal data access profile to determine a coallocation solution for allocations in memory.

30. The computer-readable medium of claim 29 wherein the temporal data access profile traces accesses of objects, and wherein the one or more data access patterns are
for object accesses.

31. The computer-readable medium of claim 29 wherein the temporal data access profile traces accesses of object fields, and wherein the one or more data access patterns are for object field accesses.

5

32. The computer-readable medium of claim 29 wherein the coallocation solution includes field reordering.

10 33. The computer-readable medium of claim 29 wherein the coallocation solution includes object splitting.

34. The computer-readable medium of claim 29 wherein the coallocation solution includes object merging.

15 35. The computer-readable medium of claim 29 wherein the coallocation solution is for guiding memory placement decisions for heap objects in subsequent execution of the computer program.

20 36. The computer-readable medium of claim 29 wherein the method further comprises altering the computer program to implement the coallocation solution.

37. The computer-readable medium of claim 36 wherein the altering comprises binary rewriting of the computer program.

25 38. The computer-readable medium of claim 36 wherein the altering comprises rewriting of part of source code of the computer program under control of a programmer.

39. The computer-readable medium of claim 36 wherein the altering comprises adjusting executable code for the computer program at compile time.

30

40. The computer-readable medium of claim 36 wherein the method further comprises:

after the altering, executing the computer program, wherein run time support software enforces the coallocation solution.

5

41. A computer-readable medium storing the computer program altered according to the method of claim 36.

42. The computer-readable medium of claim 29 wherein the temporal data access profile includes a series of data accesses spanning execution of the computer program during a profiling run.

43. The computer-readable medium of claim 29 wherein the coallocation solution is for guiding memory placement decisions for heap objects in subsequent execution of the computer program, the method further comprising:

coallocating a first set of heap objects in a first heap memory arena to improve locality for accesses of the first set of heap objects;

coallocating a second set of heap objects in a second heap memory arena to improve locality for accesses of the second set of heap objects; and

20 allocating other heap objects in a default heap memory arena.

44. A system for coallocating data in memory to improve cache performance, the system comprising:

an analysis module for determining a coallocation solution based at least in part upon a temporal data access profile of a computer program;

an instrumentation module for automating alteration of the computer program based at least in part upon the coallocation solution; and

an enforcement module for automatically enforcing the coallocation solution during execution of the altered computer program.

30

45. The system of claim 44 further comprising:
a profiling module for generating the temporal data access profile.

46. The system of claim 44 wherein the instrumentation module comprises a
5 binary re-writer.

47. The system of claim 44 wherein the instrumentation module automates
changing of heap allocation requests to heap coallocation requests.

10 48. The system of claim 44 wherein the enforcement module comprises a library
for run time support of heap coallocation requests.

49. The system of claim 44 further comprising a plurality of memory arenas in
which to implement the coallocation solution.

15 50. The system of claim 44 wherein the analysis module determines a heat value
for each of one or more hot data streams in the temporal data access profile.

51. The system of claim 44 wherein the temporal data access profile traces object
20 field accesses.

52. The system of claim 44 wherein the temporal data access profile traces object
accesses.